Particle Swarm Optimisers for Cluster formation in Wireless Sensor Networks

S. M. Guru, S. K. Halgamuge
Mech. and Manuf. Engineering, Mechatronics Research Group
The University of Melbourne, Parkville Vic 3010
Email: s.guru@pgrad.unimelb.edu.au, saman@unimelb.edu.au

S. Fernando
Ajax Engineering Fasteners
Braeside, Vic 3195 Australia
Email: sfernando@ajaxfast.com.au

Abstract

We describe the results of a performance evaluation of four extensions of Particle Swarm Optimisation (PSO) to reduce energy consumption in wireless sensor networks. Communication distances are an important factor to be reduced in sensor networks. By using clustering in a sensor network we can reduce the total communication distance, thus increasing the life of a network. We adopt a distance based clustering criterion for sensor network optimisation. From PSO perspective, we study the suitability of four different PSO algorithms for our application and propose modifications. An important modification proposed is to use a boundary checking routine for re-initialisation of a particle which moves outside the set boundary.

1. Introduction

Wireless sensor networks came into prominence with the advancement of MEMS Technology. In wireless sensor networks, sensors are expected to be battery powered tiny devices and therefore have limited energy. This makes energy consumption a critical issue in sensor networks.

In our study, we consider sensor networks in which large number of sensors are deployed. All sensors sense the environment and transmit data to a sink (destination). In cluster based communication, nodes are formed into clusters. Each cluster will have a cluster-head (CH) that will communicate with all the member nodes of a cluster. CHs transmit aggregated data to the sink. In this method, all nodes except CHs communicate short distances and only a few nodes (CHs) communicate with the sink. Clustering can greatly reduce communication cost of the nodes because they only need to send data to the nearest cluster-head. However, CH expends more energy than ordinary nodes communicating with the sink. Thus in LEACH [1], Heinzelman et al. proposed a rotation of CHs in each round of communication. In LEACH, CHs are identified by an election process influenced by randomness.

Clustering is an NP-hard problem [2]. For a given network it is always difficult to find an optimal CH placement. In LEACH-C [3], Heinzelman et al. proposed a simulated annealing approach to form clusters and identify CH positions. This method is a centralised solution assuming that the position of all nodes are known in advance and powerful computer perform the computation and inform all the nodes about their respective cluster-heads. In [4], Tillet et al. proposed a Particle Swarm Optimisation (PSO) approach for the same problem. However, the main aim was to reduce an intra-cluster distance by completely ignoring the distance to the sink. In [5], Ji et al. applied Divided Range Particle Swarm Optimisation (DRPSO) to optimise weighted clustering algorithm (WCA)[6] parameters. In contrast to the multiobjective optimisation problem, we are devicing our problem as singleobjective optimisation.

In this paper we propose an evolutionary computing based clustering method to cluster nodes uniformly distributed on a sensor field. We formulate the clustering method as a minimisation problem. We optimise the problem using four different versions of PSO. The main objective of this work is to show that:

- the choice of the specific PSO algorithm will also influence the final optimisation outcome.
- to solve application-specific problems like clustering of sensors in a sensor networks, we need to modify PSO algorithms to achieve better results.

We adopt four different Particle Swarm Optimisation methods to cluster sensor nodes in a wireless sensor network based on our clustering criterion. We have modified three algorithms to suit our application. To cluster, we consider 2 different methods:

- cluster a network when nodes have limited transmission range.
- cluster a network when nodes do not have any transmission range restrictions.

The criterion for clustering is derived based on the communication distance between nodes and the CH, and from the CH to the sink. The clustering criterion is taken from our previous work [7].

The rest of the paper is organised as follows. The brief overview of PSO and different versions of PSO used in this work is given in section 2. Section 3 summarises the assumptions and the energy model used in this paper. In Section 4, we explain the clustering methods, results and detailed analysis of the changes carried out to PSO algorithms. Finally we conclude the paper.

0-7803-9399-6/05/$20.00 © 2005 IEEE

ISSNIP 2005

2. Particle Swarm Optimisation

Particle Swarm Optimisation is an evolutionary computing technique based on principle such as bird flocking. This method was first proposed by Kennedy and Eberhart [8]. In PSO a set of potential solutions are called particles that are initialised randomly. Each particle will have a fitness value, which will be evaluated by the fitness function to be optimised in each generation. Each particle knows its best position *pbest* and the best position so far among the entire group of particles *gbest*. The particle will have velocities, which direct the flying of the particle. In each generation the velocity and the position of the particle will be updated. The velocity and the position update equations are given below as (1) and (2) respectively.

$$v_i^{k+1} = wv_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

where,

v_i^k	velocity of particle i at iteration k
v_i^{k+1}	velocity of particle i at iteration $k+1$
w	inertia weight
c_j	acceleration coefficients $j = 1, 2$
$rand_i$	random number between 0 and 1 $i = 1, 2$
s_i^k	current position of particle i at iteration k
$pbest_i$	$pbest$ of particle i
$gbest$	$gbest$ of the group
x_i^{k+1}	position of the particle i at iteration $k+1$

In recent times, there has been a number of improvements to the original PSO. We have explored different versions of PSO where the extension to the original algorithm is distinct from each other. Following PSO versions are studied in this paper:

A. PSO- Time Varying Inertia Weight (TVIW)

PSO-TVIW [9] is the basic PSO algorithm with inertia weight varying with time from 0.9 to 0.4 and the acceleration coefficient is set to 2. The time varying inertia weight is mathematically represented as follows:

$$w = (weight - 0.4) * \frac{(MAXITER - iter)}{MAXITER} + 0.4. \quad (3)$$

Where, MAXITER is the maximum iteration allowed, $iter$ is the current iteration number and $weight$ is a constant set to 0.9.

B. PSO-Time Varying Acceleration Coefficients (TVAC)

PSO-TVAC [10], proposed by Ratnaweera et al. uses time varying acceleration coefficient (TVAC). The c_1 varies from 2.5 to 0.5 and the c_2 varies from 0.5 to 2.5. Here the cognitive component is reduced and social component is increased by changing c_1 and c_2. The large cognitive component and the small social component in the initial stages of the algorithm helps the particle to wander around the search space. However, the small cognitive component and large social component at the later stages of the algorithm helps the particle to converge to the global optima. TVAC is mathematically represented as follows:

$$c_1 = (c_{1min} - c_{1max}) \frac{iter}{MAXITER} + c_{1min}, \quad (4)$$

$$c_2 = (c_{2max} - c_{2min}) \frac{iter}{MAXITER} + c_{2min}. \quad (5)$$

In Eq. 4 and 5 c_{1min} and c_{2min} are constants set to 0.5, c_{1max} and c_{2max} are also constants set to 2.5. Thus, in this algorithm as the iter progresses, c_1 varies from 2.5 to 0.5 and c_2 varies from 0.5 to 2.5.

C. Hierarchical Particle Swarm Optimizer with Time Varying Acceleration Coefficients (HPSO-TVAC)

In this method [10] the particle behaviour will not be influenced by the previous velocity term of Eq. 1. Due to non-influence of previous velocity, re-initialisation of velocity is used when the velocity stagnates in the search space. Therefore, in this method, a series of particle swarm optimisers are automatically generated inside the main particle swarm optimiser according to the behaviour of the particle in the search space, until the convergence criteria is met. The reinitialisation velocity is set proportional to V_{max}. The pseudocode for reinitialising velocity is as follows:

$v_i^{k+1} = c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k)$
  if ($v_i^{k+1} == 0$)
    if($rand_1() < 0.5$)
    v_i^{k+1} = $rand_2()$ * v
    else
    v_i^{k+1} = -$rand_3()$ * v
    end if
  endif
  v_i^{k+1} = sign(v_i^{k+1})*min(fabs(v_i^{k+1},v_{max}))

where $rand_i()$, $i = 1, 2, 3$ are separately generated uniformly distributed random numbers in the range [0,1] and v is the reinitialisation velocity. The effect of HPSO along with TVAC (hence, HPSO-TVAC) on clustering of sensor networks was observed through simulations.

D. Particle Swarm Optimisation with Supervisor-Student Model (PSO-SSM)

Liu et al. proposed PSO-SSM [11] to achieve low computational costs. The algorithm introduces a new parameter called momentum factor (mc) to update the positions of particles. In this algorithm, they also proposed a different velocity updation mechanism from the conventional PSO algorithms. Here velocity is updated only if each particle's fitness at the current iteration is not better than that of previous iteration. The velocity serves as a navigator (supervisor) by getting the right direction, while the position (student) gets a right step size along the direction. The velocity and the position are modified using the following equations.

$$v_i^{k+1} = v_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k), \quad (6)$$

$$x_i^{k+1} = (1 - mc) \times x_i^k + mc \times v_i^{k+1}. \quad (7)$$

3. Optimisation of Energy usage

When we have prior knowledge of the position of sensors deployed in a sensor network, it is always worthwhile to cluster the sensors and allow few sensors to communicate with the sink as opposed to all the sensors communicating with the sink. In this scenario, since the position of all the sensors are known, centralised algorithm should be used to cluster sensors so that always 'k' optimal clusters are formed in a network. This kind of sensor network can be used in manufacturing industries [12] where we have complete control over the deployment of sensors and the main use of the sensors are surveillance and monitoring. The main motivation is to use large number of less expensive battery operated wireless sensors instead of expensive application-specific devices. In this paper we are adopting Particle Swarm Optimisation (PSO) approach to form clusters and identify cluster-heads. We are studying the impact of the transmission range of sensor nodes and positioning of the sink in minimising the communication energy in a sensor network. Following are the assumptions we have made in this study:

- Each sensor node has fixed omni-directional transmission range.
- All nodes have identical transmission ranges and hardware configurations.
- Nodes are randomly distributed and the sensor field can be mapped into a 2-Dimensional space.
- The area of the problem space is known in advance.
- Once the nodes are deployed they are static and positions of the nodes is known to the sink beforehand. The sink runs the clustering algorithm and updates nodes about their cluster-head.
- The sensor network is densely populated with a minimum of 100 nodes in the network.

The important components of each sensor are the data and control processing unit and the radio for communication. The microprocessor used in the processing unit should be energy efficient with less energy consumption. The energy dissipation in the radio depends on the different characteristics of the radio.

A. Energy Model

The energy model used in this work is adopted from [13] [3][7] and summarised here. We have used the simple radio model to measure the energy dissipation to transmit and receive the data from a node. The energy dissipation for transmitting b bits to d distance is shown in (8). Fig. 1 depicts the parameters we are minimising in the following energy model.

$$E_{tx}(b, d) = E_{elec} \times b + \epsilon_{amp} \times b \times d^2. \quad (8)$$

The energy dissipation in a node to receive b bits of data is shown in (9)

$$E_{rx}(b) = E_{elec} \times b. \quad (9)$$

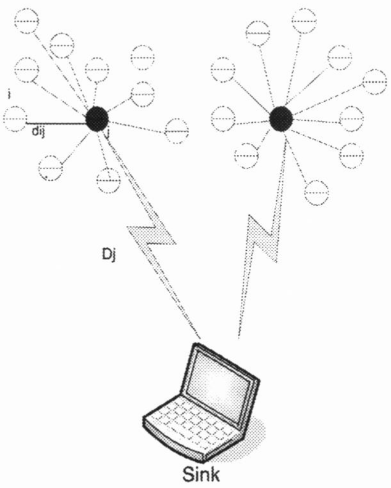

Fig. 1: Energy model based on the distance

For our experiment the transmission/receiver energy loss $E_{elec} = 50nJ/bit$ and the transmission amplifier constant is taken as $\epsilon_{amp} = 0.1nJ/bit/m^2$ [3].

Energy consumption of a wireless sensor node transmitting and receiving data from another node at a distance d can be divided into two main components: Energy used to transmit, receive and amplify data E_T and energy used for processing data E_p, mainly by the microcontroller. The total energy loss E of a sensor :

$$E = (100 + 0.1d^2)b + N_{cyc}C_{avg}V^2 + VI_0 e^{v/nv_t}\frac{N_{cyc}}{f}. \quad (10)$$

1) Energy optimisation for the network with clustered sensors: Assume a k set of sensor clusters with n_j sensors in the cluster C_j, where $1 < j <= k$. Considering the cluster C_j, in which a sensor S_{ij} is at distance d_{ij} to the cluster-head CH_j, and CH_j is at a distance of D_j to the sink. The energy loss of all the sensors and 'k' clusters can be derived using (10). Leakage current I_0 can be as large as a few mA for the microcontroller, and the effect of leakage current can be neglected for higher frequencies and lower supply voltage. Assuming the leakage current I_0 as negligible, the total energy loss for the sensor system E_{total} is:

$$\begin{aligned} E_{total} &= 0.1b\sum_{j=1}^{k}\sum_{i=1}^{n_j}(d_{ij}^2 + \frac{D_j^2}{n_j}) \\ &+(n_{total} - k)(100b + N_{cyc}C_{avg}V^2) + \\ &k(100b + N_{cyc}C_{avg}V^2)). \end{aligned} \quad (11)$$

It is clear that the first part of (11) is an energy due to distance E_{dd}. It is the only component that can be optimised independent from parameters related to the microcontroller and the supply voltage used. Consequently, E_{dd} was used as the energy loss based on the distance for formation of clusters. Please refer to [7] for detail derivation and proof.

$$E_{dd} = \sum_{j=1}^{k}\sum_{i=1}^{n_j}(d_{ij}^2 + \frac{D_j^2}{n_j}). \quad (12)$$

4. Experiment and Simulation

From (12) we can conclude that by reducing the distance from a node to the cluster-head and the cluster-head to the sink we can minimise the energy dissipation in a sensor network. Here we are not only considering the intra-cluster distance but also the distance between the cluster-head and the sink. In our simulation, we cluster the nodes taking into consideration 2 issues about sensor nodes:

- Each node can transmit or receive data from all the other nodes ($cluster_{without\_Tx\_range}$). Thus, nodes considered in this network do not have transmission range constraint.
- Nodes can transmit and receive data upto a certain distance. This is called transmission range of a node. The transmission range influences the number of clusters in a network. This method is called ($cluster_{with\_Tx\_range}$).

In ($cluster_{without\_Tx\_range}$), sensors are clustered using entirely distance based Eq.12. Here the number of clusters are user-defined hence, nodes are clustered for a given number of clusters 'k'. The fitness function for this method is as follows:

$$Fitness_{without\_Tx\_range} = \min\left(\sum_{j=1}^{k}\sum_{i=1}^{n_j}(d_{ij}^2 + \frac{D_j^2}{n_j})\right), \quad where, \sum_{j=1}^{k} n_j + k = N. \tag{13}$$

where N is the number of nodes in a network. For ($cluster_{with\_Tx\_range}$) the number of clusters for a given sensor field depends on the area of the sensor field and the transmission range of the sensors. When the node's transmission is omni-directional, limited transmission range and clusters are circular, Therefore, there will be overlapping of clusters which may lead to nodes involved in two or more clusters. To minimize overlapping of clusters we add penalty to the fitness function. We assign counters to count the number of nodes which are in the range of cluster-heads it can transmit data based on the transmission range. We classify the nodes as follows:

- n_0= number of sensors that cannot communicate with any of the CH
- n_1= number of sensors that can communicate only with a single CH
- n_2= number of sensors that can communicate with two CHs
- n_3= number of sensors that can communicate with three or more CHs

The penalty is associated with each of the above counters in the fitness function. The n_0 nodes which do not belong to any clusters will transmit data to their nearest clusterhead. The fitness function used for our simulation is:

$$Fitness_{with\_Tx\_range} = E_{dd} + E_{n_0} * (0.7*n_0 + 0.1*n_2 + 0.2*n_3) \tag{14}$$

In (14) the first term E_{dd} is the distance based communication energy of all the clusters in a sensor network. The second term is the penalty term to discourage high n_0, n_2 and n_3. The weights used for $n_i, i = 0,2,3$ were found experimentally. E_{n_0} is the summation of the squares of distances of all n_0 node to their nearest CH. In our simulations we considered a square sensor field, thus the problem of covering a sensor field by circular cluster is analogous to the number of circles that cover a square. In [14], Melissen et al. uses a simulated annealing method to cover a square in 6 and 8 circles. Since our clusters are also circular, the transmission range of each node can be equivalent to the radius of a circle. In our simulations, we have used 6 clusters. Thus, for the transmission range of a cluster-head (CH) to cover the entire square we have adopted the radius of a circle to cover a square in [14] as the transmission range of each CH to cover an entire sensor field. Therefore, in $cluster_{with\_Tx\_range}$ our objective is to group a given number of nodes into 6 clusters using Eq. 14 and the transmission range for each node is set to 29.8 units same as in [14].

A. Simulation Strategies

For our simulations, we used 100-node networks that are uniformly distributed in a 2-Dimensional problem space [0:100,0:100]. We have studied the impact of sink location on the fitness value of the PSO algorithms. In one set of simulations we considered the sink to be located remotely at (50,175). In another set of simulations we considered the sink to be located at (50,50), i.e., at the center of the network. For both simulations we use the same set of nodes and we group the nodes to form 6 clusters. We used 30 particles for our simulation and the maximum number of generations we were running was 1000. The parameters used in the simulations are tabulated in Table. 1.

Variables	Range
Population size	30
$MAXITER$	1000
v_{max}	100
x_{max}	100
v range	[0,100]
x range	[0,100]

Table 1: Initialisation and range of parameters

PSO Methods	$cluster_{without\_Tx\_range}$	$cluster_{with\_Tx\_range}$
PSO-TVIW	**25715.827**	33165.87871
PSO-TVAC	26258.274	27507.85501
HPSO-TVAC	26478.176	26382.679
PSO-SSM	25917.920	**25848.389**

Table 2: Average fitness value (10 trials) to form 6 clusters for sink located at (50,50)

PSO Methods	$cluster_{without\_Tx\_range}$	$cluster_{with\_Tx\_range}$
PSO-TVIW	**30976.103**	33690.224
PSO-TVAC	32411.009	34702.136
HPSO-TVAC	31308.336	31578.500
PSO-SSM	31213.706	**31418.503**

Table 3: Average fitness value (10 trials) to form 6 clusters for sink located at (50,175)

PSO Methods	n_0	n_1	n_2	n_3
PSO-TVIW	0.1	74	25.5	0.4
PSO-TVAC	0	66	31	2.7
HPSO-TVAC	3.9	61.4	32.2	2.5
PSO-SSM	0	69.4	30	0.6

TABLE 4: AVERAGE (10 TRIALS) NODE DISTRIBUTION IN $cluster_{with\_Tx\_range}$ WHILE FORMING 6 CLUSTERS FOR SINK LOCATED AT (50,175)

PSO Methods	n_0	n_1	n_2	n_3
PSO-TVIW	0	71.3	28.1	0.6
PSO-TVAC	0.2	63.2	34.9	1.7
HPSO-TVAC	2.8	64.9	30.3	2
PSO-SSM	0.1	68.9	30.4	0.6

TABLE 5: AVERAGE (10 TRIALS) NODES DISTRIBUTION IN $cluster_{with\_Tx\_range}$ WHILE FORMING 6 CLUSTERS FOR SINK LOCATED AT (50,50)

B. Results

We observed the performance in terms of quality of the average optimum value for 10 trials for all the PSO methods described in section 2. We ran the simulation for both $cluster_{without\_Tx\_range}$ and $cluster_{with\_Tx\_range}$ with the sink placed inside and outside the sensor field. All the PSO algorithms uses conventional method of aligning the particle with a boundary if the particle moves beyond the boundary line. However, for our application this method did not gave good results. Instead of keeping the particle at the boundary we moved a particle inside the problem space randomly. Randomness depends on the size of the problem space. Thus, there will be less possibility of a particle moving beyond the boundary in an immediate iteration. Following is the procedure for the boundary checking routine:

if ($x_{k+1} > x_{max}$)
  $x_{k+1} = 0.6 * x_{max} * rand_1 + 0.4$
else if ($x_{k+1} < x_{min}$)
  $x_{k+1} = 0.6 * x_{max} * rand_2$
end if

where $rand_i$, $i = 1, 2$ are separately generated uniformly distributed random numbers in the range [0,1]. By following the above procedure a particle would get enough space to wander around before converging to a solution when it goes beyond the boundary of a problem space. The boundary checking routine improved the convergence of a solution significantly. We used the boundary checking procedure for all the PSO methods except PSO-SSM. This is due to the ability of PSO-SSM under the influence of mc to stop particles from moving beyond the boundary of a problem space.

1) PSO-Time Varying Inertia Weight: The range of an inertia weight is varied from 0.9 to 0.5. We experiment with the acceleration co-efficient by varying and keeping it constant. There was an improvement in results with constant acceleration co-efficient of 2. This method gave best results for $cluster_{without\_Tx\_range}$. However, the performance of this algorithm was not good for $cluster_{with\_Tx\_range}$. This was largely due to a slow convergence at the initial stages of an algorithm and also compounded by the unpredictability of a fitness function.The results are shown in Table 2 and 3. Regarding the node distribution in $cluster_{with\_Tx\_range}$, more than 70% of the nodes belong to individual clusters (n_1). Nodes not belonging to any cluster were almost negligible. These results are tabulated in Table 4 and 5.

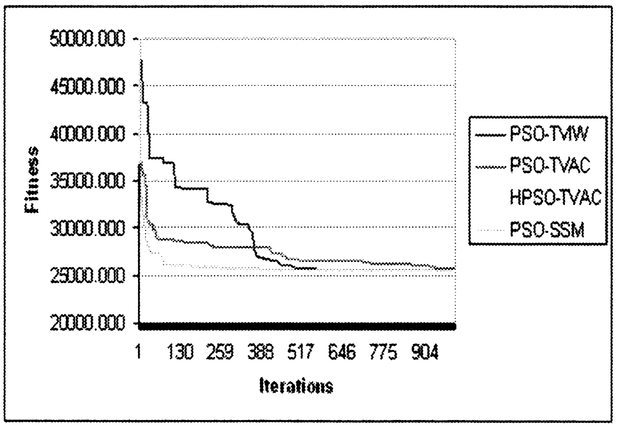

Fig. 2: convergence of solution for $cluster_{with\_Tx\_range}$ when sink was located at (50,50)

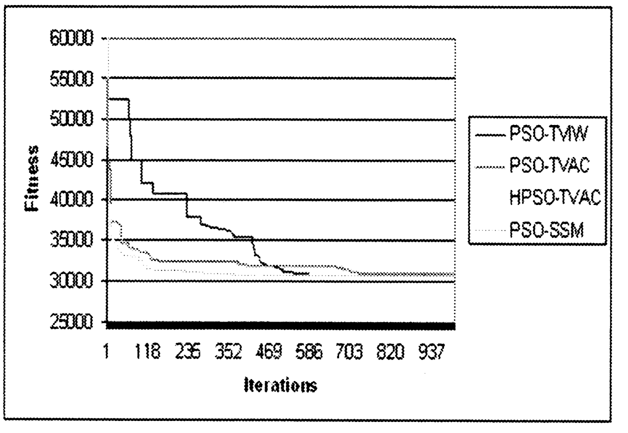

Fig. 3: convergence of solution for $cluster_{without\_Tx\_range}$ when sink was located at (50,175)

2) PSO-Time Varying Acceleration Co-efficients: In this method acceleration co-efficient c_1 was varied from 2.5 to 0.5 and c_2 was varied from 0.5 to 2.5. In [10], it was proposed to use TVIW with this method to achieve better results. However, due to the nature of our problem the performance of the fitness decreased with varying TVIW. Instead, we used less varying inertia weight $[0.5 + \frac{rand}{2}]$ as proposed in [15] with an improvement in results. Thus, we can conclude that for our application the performance of an algorithm decreases when an acceleration co-efficient and an inertia weight was both time varied. Table 5 and 4 display the node distribution for $cluster_{with\_Tx\_range}$. There are almost negligible number of nodes not belonging to any clusters and about 65% of nodes belonging to individual clusters (n_1).

3) Hierarchical Particle Swarm Optimizer with Time Varying Acceleration Coefficients (HPSO-TVAC): In our simulations, we studied the significance of re-initialisation of velocity for clustering of sensor networks. The reinitialisation velocity is the percentage of maximum velocity (v_{max}). We used $v = v_{max}$ for our simulations. Even though the algorithm results were not good for $cluster_{without\_Tx\_range}$ when compared to other PSO algorithms, from Table 2 and 3 we can conclude that the results were second best for $cluster_{with\_Tx\_range}$ and the difference between the two clustering methods fitness value was minimal. The node distribution for $cluster_{with\_Tx\_range}$ was shown in Table 4 and 5. There are around 4% and 3% of the nodes not belonging to any clusters. This would happen because of the volatile velocity. The re-initialisation of velocity will not influence the fine tuning of results. However, it would reduce solution reaching a local optima.

4) Particle Swarm Optimisation with Supervisor-Student Model (PSO-SSM): The effect of PSO-SSM method on clustering of sensor networks is reflected in Table 2 and 3. This method is second best for $cluster_{without\_Tx\_range}$ and best for $cluster_{with\_Tx\_range}$. The differences in fitness value between the clustering methods are also minimal. Table 4 and 5 displays results of the node distribution in $cluster_{with\_Tx\_range}$. We conclude that 70% of nodes belong to different individual clusters and 30% of nodes belongs to two clusters. The nodes not belonging to any clusters are almost 0.

Fig. 2 and 3 shows the convergence of all the four PSO algorithms. From the graph we can conclude that PSO-TVIW convergence is slower when compare to other algorithms. This was largely due to constant acceleration co-efficients used in the algorithm which affects the rate of convergence. This was also a reason for PSO-TVIW to perform badly for $cluster_{with\_Tx\_range}$ method. The HPSO-TVAC and PSO-SSM gave better results in $cluster_{with_Tx_range}$ when compared to other two algorithms because of the suitable mechanism to avoid local optima.

5. Conclusion

We have described the effects of different Particle Swarm Optimisers for solving the clustering problem in wireless sensor networks. We have adopted some application-specific changes to the algorithms to suit our application.

We proposed the boundary checking routine where the position of a particle will be reinitialised randomly back in the problem space instead of placing it at the boundary of the problem space. In this way frequent movement of particles out of problem space can be avoided.

We have conducted detailed simulation based experiments for the suitability of different PSO algorithms for clustering of sensor networks based on our proposed clustering methods. We have found that except PSO-SSM all the other methods need modification to produce competitive results. Even in PSO-SSM when an acceleration co-efficient varied from 0.5 to 3, the results are slightly better especially in node distribution for $cluster_{with\_Tx\_range}$ method. However, we followed the original PSO-SSM where $c_1 = c_2$ and varied from 1.5 to 6.

We have also proposed 2 clustering methods: one based on transmission range restriction of each node and the other without transmission range restriction. We conclude that the difference in the results for both clustering methods in HPSO-TVAC and PSO-SSM are minimal. We have also summarised the distance based clustering criterion used from our previous work.

Acknowledgment

The authors would like to thank Australian Research Council and Ajax Engineering Fasteners for supporting this project. We would like to thank Suhinthan Maheswararajah for discussion on PSO-SSM. We would also like to thank Madhuri Kudva for proofreading the paper.

References

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000.*, Maui, HI, 2000, pp. 3005–3014.

[2] K. P. Agarwal and P. M. C., "Exact and approximation algorithms for clustering," in *ninth annual ACM-SIAM symposium on Discrete algorithms*, San Francisco, California, U.S.A, 1998, pp. 658–667.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on*, vol. 1, no. 4, pp. 660–670, 2002.

[4] J. Tillett, R. Rao, and F. Sahin, "Cluster-head identification in ad hoc sensor networks using particle swarm optimization," in *Personal Wireless Communications, 2002 IEEE International Conference on*, 2002, pp. 201–205.

[5] C. Ji, Y. Zhang, S. Gao, P. Yuan, and Z. Li, "Particle swarm optimization for mobile ad hoc networks clustering," in *Networking, Sensing and Control, 2004 IEEE International Conference on*, vol. 1, 2004, pp. 372–375.

[6] M. Chatterjee, S. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, vol. 5, no. 2, pp. 193–204, April 2002.

[7] S. M. Guru, A. Hsu, S. Halgamuge, and S. Fernando, "An extended growing self-organising map for selection of clustering in sensor networks," *(accepted) International Journal of Distributed Sensor Networks*, vol. 1, no. 2, 2005.

[8] J. Kennedy and E. R.C., "Particle swarm optimization," in *IEEE International conference on Neural Networks*, Perth, Australia, 1995, pp. 1942–1948.

[9] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3, 1999, p. 1950 Vol. 3.

[10] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 3, pp. 240–255, 2004.

[11] Y. Liu, Z. Qin, and X. He, "Supervisor-student model in particle swarm optimization," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 1, 2004, pp. 542–547.

[12] S. M. Guru, S. Fernando, S. Halgamuge, and K. Chan, "Intelligent fastening with a-bolt technology and sensor networks," *Assembly Automation, The International Journal of assembly technology and management*, vol. 24, no. 4, pp. 386–393, 2004.

[13] A. Wang and A. Chandrakasan, "Energy-efficient dsps for wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 19, no. 4, pp. 68–78, 2002.

[14] J. Melissen and P. Schuur, "Improved coverings of a square with six and eight equal circles," 1996. [Online]. Available: citeseer.ist.psu.edu/article/melissen96improved.html

[15] Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1, 2001, pp. 81–86.